Logic and Truth Tables

Ram Chandra Dhungana, PhD

nami COLLEGE

20 May, 2024

Outline

Logical Propositions

Type of Logic

- propositional logic
- predicate logic
- fuzzy logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Following declarative sentences are propositions:

- Washington, D.C., is the capital of the United States of America.
- Toronto is the capital of Canada.
- $1 + 1 = 2$.

Logical Propositions

Type of Logic

- propositional logic
- predicate logic
- fuzzy logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Following declarative sentences are propositions:

- Washington, D.C., is the capital of the United States of America.
- Toronto is the capital of Canada.
- $1 + 1 = 2$.

Logical Propositions

Type of Logic

- propositional logic
- predicate logic
- fuzzy logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Following declarative sentences are propositions:

- Washington, D.C., is the capital of the United States of America.
- Toronto is the capital of Canada.
- $1 + 1 = 2$.

Logical Propositions

Type of Logic

- propositional logic
- predicate logic
- fuzzy logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Following declarative sentences are propositions:

- Washington, D.C., is the capital of the United States of America.
- Toronto is the capital of Canada.
- $1 + 1 = 2$.

Logical Propositions

Type of Logic

- propositional logic
- predicate logic
- fuzzy logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Following declarative sentences are propositions:

- Washington, D.C., is the capital of the United States of America.
- Toronto is the capital of Canada.
- $1 + 1 = 2$.

Logical Propositions

Type of Logic

- propositional logic
- predicate logic
- fuzzy logic

Proposition

A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Following declarative sentences are propositions:

- Washington, D.C., is the capital of the United States of America.
- Toronto is the capital of Canada.
- $1 + 1 = 2$.

Example

Following sentences are not propositions:

- What time is it?
- Read this carefully.
- x + 1 = 2.
- x + y = z.

Negation ($\neg$)

Let p be a proposition. The negation of p, denoted by $\neg p$ (also denoted by $\overline{p}$), is the statement "It is not the case that p." The proposition $\neg p$ is read "not p." The truth value of the negation of p, $\neg p$, is the opposite of the truth value of p. The truth table of negation ($\neg$) is as follows:

p	$\neg p$
T	F

Example

Following sentences are not propositions:

- What time is it?
- Read this carefully.
- x + 1 = 2.
- x + y = z.

Negation ($\neg$)

Let p be a proposition. The negation of p, denoted by $\neg p$ (also denoted by $\overline{p}$), is the statement "It is not the case that p." The proposition $\neg p$ is read "not p." The truth value of the negation of p, $\neg p$, is the opposite of the truth value of p. The truth table of negation ($\neg$) is as follows:

p	$\neg p$
T	F

Example of Negation

- The negation of "Shyam's PC runs Linux." is
  "It is not the case that Shyam's PC runs Linux."
  In other word, "Shyam's PC does not run Linux."
- The negation of "Sita's smartphone has at least 32 GB of memory" is
  "It is not the case that Sita's smartphone has at least 32 GB of memory."
  This negation can also be expressed as "Sita's smartphone does not have at least 32 GB of memory"
  or even more simply as "Sita's smartphone has less than 32 GB of memory."

Example of Negation

- The negation of "Shyam’s PC runs Linux." is
  “It is not the case that Shyam’s PC runs Linux.”
  In other word, “Shyam’s PC does not run Linux.”
- The negation of “Sita’s smartphone has at least 32 GB of memory” is
  “It is not the case that Sita’s smartphone has at least 32 GB of memory.”
  This negation can also be expressed as “Sita’s smartphone does not have at least 32 GB of memory”
  or even more simply as “Sita’s smartphone has less than 32 GB of memory.”

Example of Negation

- The negation of "Shyam's PC runs Linux." is
  "It is not the case that Shyam's PC runs Linux."
  In other word, "Shyam's PC does not run Linux."
- The negation of "Sita's smartphone has at least 32 GB of memory" is
  "It is not the case that Sita's smartphone has at least 32 GB of memory."
  This negation can also be expressed as "Sita's smartphone does not have at least 32 GB of memory"
  or even more simply as "Sita's smartphone has less than 32 GB of memory."

Example of Negation

- The negation of "Shyam's PC runs Linux." is
  "It is not the case that Shyam's PC runs Linux."
  In other word, "Shyam's PC does not run Linux."
- The negation of "Sita's smartphone has at least 32 GB of memory" is
  "It is not the case that Sita's smartphone has at least 32 GB of memory."
  This negation can also be expressed as "Sita's smartphone does not have at least 32 GB of memory"
  or even more simply as "Sita's smartphone has less than 32 GB of memory."

Example of Negation

- The negation of "Shyam's PC runs Linux." is
  "It is not the case that Shyam's PC runs Linux."
  In other word, "Shyam's PC does not run Linux."
- The negation of "Sita's smartphone has at least 32 GB of memory" is
  "It is not the case that Sita's smartphone has at least 32 GB of memory."
  This negation can also be expressed as "Sita's smartphone does not have at least 32 GB of memory"
  or even more simply as "Sita's smartphone has less than 32 GB of memory."

Example of Negation

- The negation of "Shyam's PC runs Linux." is
  "It is not the case that Shyam's PC runs Linux."
  In other word, "Shyam's PC does not run Linux."
- The negation of "Sita's smartphone has at least 32 GB of memory" is
  "It is not the case that Sita's smartphone has at least 32 GB of memory."
  This negation can also be expressed as "Sita's smartphone does not have at least 32 GB of memory"
  or even more simply as "Sita's smartphone has less than 32 GB of memory."

Logical Connective: Conjunction ($\wedge$)

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition p and q. The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Example

Find the conjunction of the propositions p and q where p is the proposition "Gita's PC has more than 16 GB free hard disk space" and q is the proposition "The processor in Gita's PC runs faster than 1 GHz." [1]

Solution: The conjunction of these propositions, $p \wedge q$, is the proposition "Gita's PC has more than 16 GB free hard disk space, and the processor in Gita's PC runs faster than 1 GHz." This conjunction can be expressed more simply as "Gita's PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz." For this conjunction to be true, both conditions given must be true. It is false when one or both of these conditions are false.

Logical Connective: Conjunction ($\wedge$)

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition p and q. The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Example

Find the conjunction of the propositions p and q where p is the proposition "Gita's PC has more than 16 GB free hard disk space" and q is the proposition "The processor in Gita's PC runs faster than 1 GHz." [1]

Solution: The conjunction of these propositions, $p \wedge q$, is the proposition "Gita's PC has more than 16 GB free hard disk space, and the processor in Gita's PC runs faster than 1 GHz." This conjunction can be expressed more simply as "Gita's PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz." For this conjunction to be true, both conditions given must be true. It is false when one or both of these conditions are false.

Logical Connective: Conjunction ($\wedge$)

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition p and q. The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Example

Find the conjunction of the propositions p and q where p is the proposition "Gita's PC has more than 16 GB free hard disk space" and q is the proposition "The processor in Gita's PC runs faster than 1 GHz." [1]

Solution: The conjunction of these propositions, $p \wedge q$, is the proposition "Gita's PC has more than 16 GB free hard disk space, and the processor in Gita's PC runs faster than 1 GHz." This conjunction can be expressed more simply as "Gita's PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz." For this conjunction to be true, both conditions given must be true. It is false when one or both of these conditions are false.

Logical Connective: Disjunction, Inclusive Or ($\vee$)

Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q." The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Example

Translate the statement "Students who have taken calculus or introductory computer science can take this class" in a statement in propositional logic using the propositions p: "A student who has taken calculus can take this class" and q: "A student who has taken introductory computer science can take this class."

Solution: We assume that this statement means that students who have taken both calculus

Logical Connective: Disjunction, Inclusive Or ($\vee$)

Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q." The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Example

Translate the statement "Students who have taken calculus or introductory computer science can take this class" in a statement in propositional logic using the propositions p: "A student who has taken calculus can take this class" and q: "A student who has taken introductory computer science can take this class."

Solution: We assume that this statement means that students who have taken both calculus

Exclusive Or ($\oplus$)

Let p and q be propositions. The exclusive or of p and q, denoted by $p \oplus q$ (or p XOR q), is the proposition that is true when exactly one of p and q is true and is false otherwise.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Example

Let p and q be the propositions that state "A student can have a salad with dinner" and "A student can have soup with dinner," respectively. What is $p \oplus q$, the exclusive or of p and q?
Solution: The exclusive or of p and q is the statement that is true when exactly one of p and q is true. That is, $p \oplus q$ is the statement "A student can have soup or salad, but not both, with dinner." Note that this is often stated as "A student can have soup or a salad with dinner," without explicitly stating that taking both is not permitted.

Conditional Statement: If then ($\rightarrow$)

Let p and q be propositions. The conditional statement $p \rightarrow q$ is the proposition "if p, then q." The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

The statement $p \rightarrow q$ is called a conditional statement because $p \rightarrow q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

Example

- the bulb lights $\rightarrow$ the power supply is connected Which may be read as:
  The bulb may not light but, if it does, then the power supply must be connected.

- Let p be the statement “Maria learns discrete mathematics” and q the statement “Maria will find a good job.” Express the statement $p \rightarrow q$ as a statement in English.
  Solution: From the definition of conditional statements, we see that when p is the statement “Maria learns discrete mathematics” and q is the statement “Maria will find a good job,” $p \rightarrow q$ represents the statement
  “If Maria learns discrete mathematics, then she will find a good job.”
  There are many other ways to express this conditional statement in English. Among the most natural of these are

Classwork

1. Show that $\vee$ is a commutative operator, i.e. $p \vee q \equiv q \vee p$.
2. Show that $\wedge$ is a commutative operator, i.e. $p \wedge q \equiv q \wedge p$.
3. Show that $\vee$ is associative operator, i.e. $p \vee (q \vee r) \equiv (p \vee q) \vee r$.
4. Show that $\wedge$ is associative operator, i.e. $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$.
5. $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$, shows $\wedge$ is a distributive operator.
6. $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$, shows $\vee$ is a distributive operator.
7. $\neg(p \wedge q) \equiv (\neg p) \vee (\neg q)$, De Morgan's Law.
8. $\neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$, De Morgan's Law.
9. $\neg(\neg p) \equiv p$, double negation property.

Converse, Contrapositive, and Inverse

Converse

The proposition $q \to p$ is called the converse of $p \to q$.

Contrapositive

The proposition is the contrapositive of $p \to q$ is the proposition $\neg q \to \neg p$.

Inverse

The proposition $\neg p \to \neg q$ is called the inverse of $p \to q$.

Equivalent

When two compound propositions always have the same truth values, regardless of the truth values of its propositional variables, we call them equivalent.

Converse, Contrapositive, and Inverse

Converse

The proposition $q \to p$ is called the converse of $p \to q$.

Contrapositive

The proposition is the contrapositive of $p \to q$ is the proposition $\neg q \to \neg p$.

Inverse

The proposition $\neg p \to \neg q$ is called the inverse of $p \to q$.

Equivalent

When two compound propositions always have the same truth values, regardless of the truth values of its propositional variables, we call them equivalent.

Converse, Contrapositive, and Inverse

Converse

The proposition $q \to p$ is called the converse of $p \to q$.

Contrapositive

The proposition is the contrapositive of $p \to q$ is the proposition $\neg q \to \neg p$.

Inverse

The proposition $\neg p \to \neg q$ is called the inverse of $p \to q$.

Equivalent

When two compound propositions always have the same truth values, regardless of the truth values of its propositional variables, we call them equivalent.

Converse, Contrapositive, and Inverse

Converse

The proposition $q \to p$ is called the converse of $p \to q$.

Contrapositive

The proposition is the contrapositive of $p \to q$ is the proposition $\neg q \to \neg p$.

Inverse

The proposition $\neg p \to \neg q$ is called the inverse of $p \to q$.

Equivalent

When two compound propositions always have the same truth values, regardless of the truth values of its propositional variables, we call them equivalent.

Truth table of Converse, Contrapositive, and Inverse

p	q	$\neg p$	$\neg q$	Statement $p \rightarrow q$	Converse $q \rightarrow p$	Contrapositive $\neg q \rightarrow \neg p$	Invers $\neg p \rightarrow$
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	T
F	T	T	F	T	F	T	F
F	F	T	T	T	T	T	T

Note

A conditional statement and its contrapositive are equivalent. The converse and the inverse of a conditional statement are also equivalent, but neither is equivalent to the original conditional statement.

Example

Find the contrapositive, the converse, and the inverse of the conditional statement:

"The home team wins whenever it is raining."

Solution: Because "q whenever p" is one of the ways to express the conditional statement $p \to q$, the original statement can be rewritten as

"If it is raining, then the home team wins."

Consequently, the contrapositive of this conditional statement is

"If the home team does not win, then it is not raining."

The converse is

"If the home team wins, then it is raining."

The inverse is

"If it is not raining, then the home team does not win."

Biconditional

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

There are some other common ways to express $p \leftrightarrow q$:

"p is necessary and sufficient for q"
"if p then q, and conversely"
"p iff q." "p exactly when q."

Biconditional

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition “p if and only if q.” The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

There are some other common ways to express $p \leftrightarrow q$:

“p is necessary and sufficient for q”
“if p then q, and conversely”
“p iff q.” “p exactly when q.”

Biconditional

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

There are some other common ways to express $p \leftrightarrow q$:

"p is necessary and sufficient for q"
"if p then q, and conversely"
"p iff q." "p exactly when q."

Example

Let p be the statement “You can take the flight,” and let q be the statement “You buy a ticket.” Then $p \leftrightarrow q$ is the statement “You can take the flight if and only if you buy a ticket.”

Example

$valid\_user\_number \wedge matching\_password \leftrightarrow login\_successful$

Example

Let p be the statement "You can take the flight," and let q be the statement "You buy a ticket." Then $p \leftrightarrow q$ is the statement "You can take the flight if and only if you buy a ticket."

Example

$valid\_user\_number \wedge matching\_password \leftrightarrow login\_successful$

Aircraft

An aircraft has a fixed capacity and it is required to record the number of people aboard the aircraft at any time. The aircraft seats are not numbered and passengers enter the aircraft and choose seats on a first-come-first-served basis.

- Passengers belong to the set of all possible persons. Let this set be called PERSON.
- Hence, for this system, the basic type (or given set) is:
  [PERSON]: the set of all possible uniquely identified persons.
- If capacity denotes the seating capacity of the aircraft we have:
  capacity : $\mathbb{N}$- the seating capacity of the aircraft
- At any time the state of the system is given by the number of passengers on the aircraft. We can describe this state by a set of persons, onboard (which will be one of the many

Given Information

[PERSON]	The set of all possible uniquely identified
capacity : $\mathbb{N}$	The seating capacity of the aircraft
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft (one of the many possible subsets of PER
#onboard $\leq$ capacity	Constraint which is an invariant for syster
onboard = { }	Initial state of the system

Boarding operation

The boarding operation may be defined by

p : PERSON	p is a person
$p \notin$ onboard	Precondition for embarkation
# onboard < capacity	Precondition for embarkation
onboard' = onboard $\cup$ {p}	True after p embarks on aircraft

Disembarkation

Disembarkation is specified by:

p : PERSON	p is a person
$p \in$ onboard	Precondition for disembarkation
onboard' = onboard $\setminus$ {p}	True after p disembarks

Boarding operation

The boarding operation may be defined by

$p : PERSON$	p is a person
$p \notin onboard$	Precondition for embarkation
$\# onboard < capacity$	Precondition for embarkation
$onboard' = onboard \cup \{p\}$	True after p embarks on aircraft

Disembarkation

Disembarkation is specified by:

$p : PERSON$	p is a person
$p \in onboard$	Precondition for disembarkation
$onboard' = onboard \setminus \{p\}$	True after p disembarks

Total system

[PERSON]	The set of all possible uniquely identified persons
capacity : $\mathbb{N}$	The seating capacity of the aircraft
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for system
onboard = { }	Initial state of the system
p : PERSON	
reply : FEEDBACK	

{ (p $\notin$ onboard $\wedge$ # onboard $<$ capacity $\wedge$ onboard' = onboard $\cup$ {p} $\wedge$ reply = OK)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard = capacity $\wedge$ onboard' = onboard $\wedge$ reply = two errors)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard $<$ capacity $\wedge$ onboard' = onboard $\wedge$ reply = on board)

$\oplus$

(p $\notin$ onboard $\wedge$ #onboard = capacity $\wedge$ onboard' = onboard $\wedge$ reply = full) }

Total system

[PERSON]	The set of all possible uniquely identified persons
capacity : $\mathbb{N}$	The seating capacity of the aircraft
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for system
onboard = { }	Initial state of the system
p : PERSON	
reply : FEEDBACK	

{ (p $\notin$ onboard $\wedge$ # onboard < capacity $\wedge$ onboard' = onboard $\cup$ {p} $\wedge$ reply = OK)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard = capacity $\wedge$ onboard' = onboard $\wedge$ reply = two errors)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard < capacity $\wedge$ onboard' = onboard $\wedge$ reply = on board)

$\oplus$

(p $\notin$ onboard $\wedge$ #onboard = capacity $\wedge$ onboard' = onboard $\wedge$ reply = full) }

Total system

[PERSON]	The set of all possible uniquely identified persons
capacity : $\mathbb{N}$	The seating capacity of the aircraft
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for system
onboard = { }	Initial state of the system

p : PERSON

reply : FEEDBACK

{ (p $\notin$ onboard $\wedge$ # onboard < capacity $\wedge$ onboard' = onboard $\cup$ {p} $\wedge$ reply = OK)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard = capacity $\wedge$ onboard' = onboard $\wedge$ reply = two errors)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard < capacity $\wedge$ onboard' = onboard $\wedge$ reply = on board)

$\oplus$

(p $\notin$ onboard $\wedge$ #onboard = capacity $\wedge$ onboard' = onboard $\wedge$ reply = full) }

Total system

[PERSON]	The set of all possible uniquely identified persons
capacity : $\mathbb{N}$	The seating capacity of the aircraft
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for system
onboard = { }	Initial state of the system

p : PERSON

reply : FEEDBACK

{ (p $\notin$ onboard $\wedge$ # onboard < capacity $\wedge$ onboard’ = onboard $\cup$ {p} $\wedge$ reply = OK)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard = capacity $\wedge$ onboard’ = onboard $\wedge$ reply = two errors)

$\oplus$

(p $\in$ onboard $\wedge$ # onboard < capacity $\wedge$ onboard’ = onboard $\wedge$ reply = on board)

$\oplus$

(p $\notin$ onboard $\wedge$ #onboard = capacity $\wedge$ onboard’ = onboard $\wedge$ reply = full) }

Total system

[PERSON]	The set of all possible uniquely identified persons
capacity : $\mathbb{N}$	The seating capacity of the aircraft
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for system
onboard = { }	Initial state of the system

p : PERSON
reply : FEEDBACK

$$\{\ (p \notin \text{onboard} \wedge \#\ \text{onboard} < \text{capacity} \wedge \text{onboard'} = \text{onboard} \cup \{p\} \wedge \text{reply} = \text{OK})$$

$$\oplus$$

$$(\ p \in \text{onboard} \wedge \#\ \text{onboard} = \text{capacity} \wedge \text{onboard'} = \text{onboard} \wedge \text{reply} = \text{two errors})$$

$$\oplus$$

$$(\ p \in \text{onboard} \wedge \#\ \text{onboard} < \text{capacity} \wedge \text{onboard'} = \text{onboard} \wedge \text{reply} = \text{on board})$$

$$\oplus$$

$$(p \notin \text{onboard} \wedge \#\text{onboard} = \text{capacity} \wedge \text{onboard'} = \text{onboard} \wedge \text{reply} = \text{full})\ \}$$

Disembarkation

[PERSON]	The set of all possible uniquely identified
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for syster
onboard = { }	Initial state of the system
p : PERSON	
reply : FEEDBACK	

$$\{\ (p \in \text{onboard} \wedge \text{onboard'} = \text{onboard} \setminus \{p\} \wedge \text{reply} = \text{OK})$$

$$\oplus$$

$$(p \notin \text{onboard} \wedge \text{onboard'} = \text{onboard} \wedge \text{reply} = \text{not on board})\}$$

Disembarkation

[PERSON]	The set of all possible uniquely identified
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for syster
onboard = { }	Initial state of the system

p : PERSON

reply : FEEDBACK

$$\{\, (p \in \text{onboard} \wedge \text{onboard}' = \text{onboard} \setminus \{p\} \wedge \text{reply} = \text{OK})$$

$$\oplus$$

$$(p \notin \text{onboard} \wedge \text{onboard}' = \text{onboard} \wedge \text{reply} = \text{not on board})\}$$

Disembarkation

[PERSON]	The set of all possible uniquely identified
onboard : $\mathbb{P}$ PERSON	The set of persons on the aircraft
onboard $\leq$ capacity	Constraint which is an invariant for syster
onboard = { }	Initial state of the system

p : PERSON

reply : FEEDBACK

$$\{\ (p \in \text{onboard} \wedge \text{onboard'} = \text{onboard} \setminus \{p\} \wedge \text{reply} = \text{OK})$$

$$\bigoplus$$

$$(p \notin \text{onboard} \wedge \text{onboard'} = \text{onboard} \wedge \text{reply} = \text{not on board})\}$$

Assignment

A college provides a multi-user computer system for its members. All members must register with the college's IT Services unit before they are allowed access to the computer system. To use the system, each registered user must *log_in*. At any given time a registered user will either be logged-in or not logged-in and it is not possible for a user to be logged-in more than once concurrently. In the following, express your solutions both symbolically (using sets and propositional logic), in the style outlined towards the end of the preceding notes, and in narrative form using plain English. Your solutions should cater for the necessary preconditions not being satisfied and be based upon a free type of the form:

$$RESPONSE ::= OK \mid Already\_a\_user \mid Not\_a\_user \mid Logged\_in \mid Not\_logged\_in$$

1. Define an operation to register a new user.
2. Define an operation to cancel a user's registration.
3. Define an operation to log-in.
4. Define an operation to log-out.

Classwork

Let P, Q and R be the propositions:

P: the membership is less than 20

Q: all the members are men

R: the maximum number of members is 50

1. Describe, in English, the meaning of the following propositions:
   a. $P \wedge Q$
   b. $\neg R$
2. Using P, Q and R as defined, represent, symbolically, the proposition: There are at least 20 members and some of them are women

Classwork

Suppose P represents the proposition “Claire is happy” and Q represents the proposition “Claire is rich”. Write, in symbolic form:

1. Claire is poor but happy
2. Claire is neither rich nor happy
3. Claire is either rich or unhappy
4. Claire is either poor or is both rich and unhappy.

Classwork

If P, Q, R and S represent logical propositions:

1. Without altering their essential meaning, simplify the following predicates by omitting as many parentheses as possible:
   a. $(\neg(((\neg R) \wedge P) \vee Q)) \rightarrow R$
   b. $(\neg P) \rightarrow (((P \rightarrow Q) \rightarrow R) \wedge S)$
2. Insert parentheses to emphasise how the following predicates are interpreted according to the conventional precedence rules:
   a. $P \rightarrow Q \leftrightarrow \neg Q \rightarrow \neg P$
   b. $P \vee Q \wedge \neg R \vee Q \wedge P$
3. Any logical proposition which is always true is called a tautology; any which is always false is called a contradiction. Identify which of the following propositions is a tautology, a contradiction or neither:
   a. $\neg(P \vee Q)$
   b. $false \wedge \neg(P \vee Q)$
   c. $false \vee true$

Classwork

If P, Q, R and S represent logical propositions:

1. Without altering their essential meaning, simplify the following predicates by omitting as many parentheses as possible:
   a. $(\neg(((\neg R) \wedge P) \vee Q)) \to R$
   b. $(\neg P) \to (((P \to Q) \to R) \wedge S)$
2. Insert parentheses to emphasise how the following predicates are interpreted according to the conventional precedence rules:
   a. $P \to Q \leftrightarrow \neg Q \to \neg P$
   b. $P \vee Q \wedge \neg R \vee Q \wedge P$
3. Any logical proposition which is always true is called a tautology; any which is always false is called a contradiction. Identify which of the following propositions is a tautology, a contradiction or neither:
   a. $\neg(P \vee Q)$
   b. $\mathit{false} \wedge \neg(P \vee Q)$
   c. $\mathit{false} \vee \mathit{true}$

Classwork

Consider the following information: Oscar either cycles to work or uses his car. If it is not raining, Oscar cycles to work. If it is raining then Oscar uses his car unless the car does not start, in which case he has to cycle to work in the rain unless he can get a push-start from his neighbour. If P, Q and R represent propositions as follows:

	P :	a push-start is available;
propositions as follows:	Q :	the car starts;
	R :	it is raining

Write down a logical expression involving P, Q and R which evaluates to true if Oscar cycles to work and false otherwise.

Classwork

All recognized modules which students at a university can study are modelled by the set modules. Modules that are taken in the first year are modelled by the set firstYear; those that are taken in the second year by the set secondYear; and those in the third year by the set thirdYear. Express each of the following statements using set notation (note that the statements may not be consistent with each other).

1. The total number of recognized modules that are available to be studied will never exceed 50.
2. None of the available modules can be taken in different years of a course (i.e. every module can be taken only in the first year or only in the second year or only in the third year - where “or” is exclusive).
3. The module computing_fundamentals is taught in the first year.
4. The computer_architecture module may be taught in either year two or year three but never in both years.
5. All modules taken in years one, two or three are recognized by the university.

Predicate Logic

Statements involving variables, such as
"x > 3," "x = y + 3," "x + y = z," and
"Computer x is under attack by an intruder,"
and
"Computer x is functioning properly,"

These statements are neither true nor false when the values of the variables are not specified.

The statement "x is greater than 3" has two parts. The first part, the variable x, is the subject of the statement. The second part—the predicate, "is greater than 3"—refers to a property that the subject of the statement can have. We can denote the statement "x is greater than 3" by P(x), where P denotes the predicate "is greater than 3" and x is the variable. The statement P(x) is also said to be the value of the propositional function P at x. Once a value has been assigned to the variable x, the statement P(x) becomes a proposition and has a truth

Predicate Logic

Statements involving variables, such as
"x > 3," "x = y + 3," "x + y = z," and
"Computer x is under attack by an intruder,"
and
"Computer x is functioning properly,"

These statements are neither true nor false when the values of the variables are not specified.

The statement "x is greater than 3" has two parts. The first part, the variable x, is the subject of the statement. The second part—the predicate, "is greater than 3"—refers to a property that the subject of the statement can have. We can denote the statement "x is greater than 3" by P(x), where P denotes the predicate "is greater than 3" and x is the variable. The statement P(x) is also said to be the value of the propositional function P at x. Once a value has been assigned to the variable x, the statement P(x) becomes a proposition and has a truth

Example

Let P(x) denote the statement “x > 3.” What are the truth values of P(4) and P(2)?
Solution: We obtain the statement P(4) by setting x = 4 in the statement “x > 3.” Hence, P(4), which is the statement “4 > 3,” is true. However, P(2), which is the statement “2 > 3,” is false.

Example

Let Q(x, y) denote the statement “x = y + 3.” What are the truth values of the propositions Q(1, 2) and Q(3, 0)?
Solution: To obtain Q(1, 2), set x = 1 and y = 2 in the statement Q(x, y). Hence, Q(1, 2) is the statement “1 = 2 + 3,” which is false. The statement Q(3, 0) is the proposition “3 = 0 + 3,” which is true.

Example

Let $P(x)$ denote the statement "$x > 3$." What are the truth values of $P(4)$ and $P(2)$?

Solution: We obtain the statement $P(4)$ by setting $x = 4$ in the statement "$x > 3$." Hence, $P(4)$, which is the statement "$4 > 3$," is true. However, $P(2)$, which is the statement "$2 > 3$," is false.

Example

Let $Q(x, y)$ denote the statement "$x = y + 3$." What are the truth values of the propositions $Q(1, 2)$ and $Q(3, 0)$?

Solution: To obtain $Q(1, 2)$, set $x = 1$ and $y = 2$ in the statement $Q(x, y)$. Hence, $Q(1, 2)$ is the statement "$1 = 2 + 3$," which is false. The statement $Q(3, 0)$ is the proposition "$3 = 0 + 3$," which is true.

Quantifiers

When the variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value. However, there is another important way, called quantification, to create a proposition from a propositional function. Quantification expresses the extent to which a predicate is true over a range of elements. In English, the words all, some, many, none, and few are used in quantifications. We will focus on two types of quantification here: **universal quantification**, which tells us that a predicate is true for every element under consideration, and **existential quantification**, which tells us that there is one or more element under consideration for which the predicate is true. The area of logic that deals with predicates and quantifiers is called the **predicate calculus**.

The Universal Quantifier

The universal quantification of P(x) is the statement

"P(x) for all values of x in the domain."

The notation $\forall xP(x)$ denotes the universal quantification of P(x). Here $\forall$ is called the universal quantifier. We read $\forall xP(x)$ as "for all xP(x)" or "for every xP(x)." An element for which P(x) is false is called a counterexample to $\forall\ xP(x)$

Example

Let P(x) be the statement "$x + 1 > x$." What is the truth value of the quantification $\forall xP(x)$, where the domain consists of all real numbers?
Solution: Because P(x) is true for all real numbers x, the quantification $\forall xP(x)$ is true.

Example

Let Q(x) be the statement "$x < 2$." What is the truth value of the quantification $\forall xQ(x)$, where the domain consists of all real numbers?
Solution: Q(x) is not true for every real number x, because, for instance, Q(3) is false. That is, $x = 3$ is a counterexample for the statement $\forall xQ(x)$. Thus, $\forall xQ(x)$ is false.

The Existential Quantifier

The existential quantification of P(x) is the proposition "There exists an element x in the domain such that P(x)." We use the notation $\exists\, xP(x)$ for the existential quantification of P(x). Here $\exists$ is called the existential quantifier.

A domain must always be specified when a statement $\exists xP(x)$ is used. Furthermore, the meaning of $\exists xP(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists xP(x)$ has no meaning. Besides the phrase "there exists," we can also express existential quantification in many other ways, such as by using the words

"for some," "for at least one," or "there is." The existential quantification $\exists xP(x)$ is read as
"There is an x such that P(x),"
"There is at least one x such that P(x),"
or

The Existential Quantifier

The existential quantification of P(x) is the proposition "There exists an element x in the domain such that P(x)." We use the notation $\exists\, xP(x)$ for the existential quantification of P(x). Here $\exists$ is called the existential quantifier. A domain must always be specified when a statement $\exists xP(x)$ is used. Furthermore, the meaning of $\exists xP(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists xP(x)$ has no meaning. Besides the phrase "there exists," we can also express existential quantification in many other ways, such as by using the words

"for some," "for at least one," or "there is." The existential quantification $\exists xP(x)$ is read as
"There is an x such that P(x),"
"There is at least one x such that P(x),"
or

Example

Suppose that P(x) is "$x^2 > 0$." To show that the statement $\forall$ xP(x) is false where the universe of discourse consists of all integers, we give a counterexample. We see that $x = 0$ is a counterexample because $x^2 = 0$ when $x = 0$, so that x^2 is not greater than 0 when $x = 0$.

Example

What does the statement $\forall$xN(x) mean if N(x) is "Computer x is connected to the network" and the domain consists of all computers on campus?
Solution: The statement $\forall$xN(x) means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as "Every computer on campus is connected to the network."

Example

Suppose that P(x) is "$x^2 > 0$." To show that the statement ∀ xP(x) is false where the universe of discourse consists of all integers, we give a counterexample. We see that $x = 0$ is a counterexample because $x^2 = 0$ when $x = 0$, so that x^2 is not greater than 0 when $x = 0$.

Example

What does the statement ∀xN(x) mean if N(x) is "Computer x is connected to the network" and the domain consists of all computers on campus?

Solution: The statement ∀xN(x) means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as "Every computer on campus is connected to the network."

Example

Suppose that P(x) is "$x^2 > 0$." To show that the statement $\forall$ xP(x) is false where the universe of discourse consists of all integers, we give a counterexample. We see that $x = 0$ is a counterexample because $x^2 = 0$ when $x = 0$, so that x^2 is not greater than 0 when $x = 0$.

Example

What does the statement ∀xN(x) mean if N(x) is "Computer x is connected to the network" and the domain consists of all computers on campus?

Solution: The statement ∀xN(x) means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as "Every computer on campus is connected to the network."

Quantifiers

The Uniqueness Quantifier

There is no limitation on the number of different quantifiers we can define, such as “there are exactly two,” “there are no more than three,” “there are at least 100,” and so on. Of these other quantifiers, the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$, $\exists 1$. The notation $\exists! x P(x)$ [or $\exists 1 x P(x)$] states “There exists a unique x such that P(x) is true.”

Example

For instance, $\exists! x(x - 1 = 0)$, where the domain is the set of real numbers, states that there is a unique real number x such that $x - 1 = 0$. This is a true statement, as $x = 1$ is the unique real number such that $x - 1 = 0$. Observe that we can use quantifiers and propositional logic to express uniqueness, so the uniqueness quantifier can be avoided.

The Uniqueness Quantifier

There is no limitation on the number of different quantifiers we can define, such as “there are exactly two,” “there are no more than three,” “there are at least 100,” and so on. Of these other quantifiers, the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$, $\exists 1$. The notation $\exists! x P(x)$ [or $\exists 1 x P(x)$] states “There exists a unique x such that P(x) is true.”

Example

For instance, $\exists! x(x - 1 = 0)$, where the domain is the set of real numbers, states that there is a unique real number x such that $x - 1 = 0$. This is a true statement, as $x = 1$ is the unique real number such that $x - 1 = 0$. Observe that we can use quantifiers and propositional logic to express uniqueness, so the uniqueness quantifier can be avoided.

Quantification Over Finite Domain

When the domain of a quantifier is finite, that is, when all its elements can be listed, quantified statements can be expressed using propositional logic. In particular, when the elements of the domain are $x_1, x_2, \ldots, x_n$, where n is a positive integer, the universal quantification ∀xP(x) is the same as the conjunction

$$P(x_1) \wedge P(x2) \wedge \cdots \wedge P(x_n),$$

because this conjunction is true if and only if $P(x_1), P(x_2), \ldots, P(x_n)$ are all true.

Example

Real life example???

Quantification Over Finite Domain

When the domain of a quantifier is finite, that is, when all its elements can be listed, quantified statements can be expressed using propositional logic. In particular, when the elements of the domain are $x_1, x_2, \ldots, x_n$, where n is a positive integer, the universal quantification ∀xP(x) is the same as the conjunction

$$P(x_1) \wedge P(x2) \wedge \cdots \wedge P(x_n),$$

because this conjunction is true if and only if $P(x_1), P(x_2), \ldots, P(x_n)$ are all true.

Example

Real life example???

Example

What is the truth value of $\forall x P(x)$, where P(x) is the statement "$x^2 < 10$" and the domain consists of the positive integers not exceeding 4?

Solution: The statement ∀xP(x) is the same as the conjunction

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4)$$

, because the domain consists of the integers $1, 2, 3$, and 4. Because P(4), which is the statement "$4^2 < 10$", is false, it follows that ∀xP(x) is false.

Connections Quantification and Looping

To determine whether $\forall x P(x)$ is true, we can loop through all n values of x to see whether P(x) is always true. If we encounter a value x for which P(x) is false, then we have shown that $\forall x P(x)$ is false. Otherwise, $\forall x P(x)$ is true. To see whether $\exists x P(x)$ is true, we loop through the n values of x searching for a value for which P(x) is true. If we find one, then $\exists x P(x)$ is true. If we never find such an x, then we have determined that $\exists x P(x)$ is false.
(Note that this searching procedure does not apply if there are infinitely many values in the domain. However, it is still a useful way of thinking about the truth values of quantifications.)

Connections Quantification and Looping

To determine whether $\forall x P(x)$ is true, we can loop through all n values of x to see whether P(x) is always true. If we encounter a value x for which P(x) is false, then we have shown that $\forall x P(x)$ is false. Otherwise, $\forall x P(x)$ is true. To see whether $\exists x P(x)$ is true, we loop through the n values of x searching for a value for which P(x) is true. If we find one, then $\exists x P(x)$ is true. If we never find such an x, then we have determined that $\exists x P(x)$ is false. **(Note that this searching procedure does not apply if there are infinitely many values in the domain. However, it is still a useful way of thinking about the truth values of quantifications.)**

Quantifiers with Restricted Domains

Example

What do the statements $\forall x < 0(x^2 > 0), \forall y \neq 0(y^3 \neq 0)$, and $\exists z > 0(z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

Solution: The statement $\forall x < 0(x^2 > 0)$ states that for every real number x with $x < 0, x^2 > 0$. That is, it states "The square of a negative real number is positive." This statement is the same as

$$\forall x(x < 0 \rightarrow x^2 > 0).$$

The statement $\forall y \neq 0(y^3 \neq 0)$ states that for every real number y with $y \neq 0$, we have $y^3 \neq 0$. That is, it states "The cube of every nonzero real number is nonzero." This statement is equivalent to

$$\forall y(y \neq 0 \rightarrow y^3 \neq 0).$$

Binding Variables

When a quantifier is used on the variable x, we say that this occurrence of the variable is **bound**. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be **free**. All the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition. This can be done using a combination of universal quantifiers, existential quantifiers, and value assignments. The part of a logical expression to which a quantifier is applied is called the scope of this quantifier. Consequently, a variable is free if it is outside the scope of all quantifiers in the formula that specify this variable.

Example

In the statement $\exists x(x + y = 1)$, the variable x is bound by the existential quantification $\exists x$, but the variable y is free because it

Binding Variables

When a quantifier is used on the variable x, we say that this occurrence of the variable is **bound**. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be **free**. All the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition. This can be done using a combination of universal quantifiers, existential quantifiers, and value assignments. The part of a logical expression to which a quantifier is applied is called the scope of this quantifier. Consequently, a variable is free if it is outside the scope of all quantifiers in the formula that specify this variable.

Example

In the statement $\exists x(x + y = 1)$, the variable x is bound by the existential quantification $\exists x$, but the variable y is free because it

Logical Equivalences Involving Quantifiers

Statements involving predicates and quantifiers are logically equivalent if and only if they have the same truth value no matter which predicates are substituted into these statements and which domain of discourse is used for the variables in these propositional functions. We use the notation $S \equiv T$ to indicate that two statements S and T involving predicates and quantifiers are logically equivalent.

Example

The statements $\forall x(P(x) \wedge Q(x))$ and $\forall x P(x) \wedge \forall x Q(x)$ are logically equivalent (where the same domain is used throughout). This logical equivalence shows that we can distribute a universal quantifier over a conjunction. Furthermore, we can also distribute an existential quantifier over a disjunction. However, we cannot distribute a universal quantifier over a

Logical Equivalences Involving Quantifiers

Statements involving predicates and quantifiers are logically equivalent if and only if they have the same truth value no matter which predicates are substituted into these statements and which domain of discourse is used for the variables in these propositional functions. We use the notation $S \equiv T$ to indicate that two statements S and T involving predicates and quantifiers are logically equivalent.

Example

The statements $\forall x(P(x) \wedge Q(x))$ and $\forall xP(x) \wedge \forall xQ(x)$ are logically equivalent (where the same domain is used throughout). This logical equivalence shows that we can distribute a universal quantifier over a conjunction. Furthermore, we can also distribute an existential quantifier over a disjunction. However, we cannot distribute a universal quantifier over a

Negating Quantified Expressions

When the domain of a predicate $P(x)$ consists of n elements, where n is a positive integer greater than one, the rules for negating quantified statements are exactly the same as De Morgan's laws discussed in Section 1.3. This is why these rules are called De Morgan's laws for quantifiers. When the domain has n elements $x_1, x_2, \ldots, x_n$, it follows that $\neg\forall x P(x)$ is the same as $\neg(P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n))$, which is equivalent to $\neg P(x_1) \vee \neg P(x_2) \vee \cdots \vee \neg P(x_n)$ by De Morgan's laws, and this is the same as $\exists x \neg P(x)$.

Classwork

Show that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Solution: By De Morgan's law for universal quantifiers, we know that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(\neg(P(x) \rightarrow Q(x)))$ are logically equivalent. By logical equivalence in Table 7 we know that $\neg(P(x) \rightarrow Q(x))$ and

Negating Quantified Expressions

When the domain of a predicate $P(x)$ consists of n elements, where n is a positive integer greater than one, the rules for negating quantified statements are exactly the same as De Morgan's laws discussed in Section 1.3. This is why these rules are called De Morgan's laws for quantifiers. When the domain has n elements $x_1, x_2, \ldots, x_n$, it follows that $\neg\forall x P(x)$ is the same as $\neg(P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n))$, which is equivalent to $\neg P(x_1) \vee \neg P(x_2) \vee \cdots \vee \neg P(x_n)$ by De Morgan's laws, and this is the same as $\exists x \neg P(x)$.

Classwork

Show that $\neg\forall x(P(x) \to Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Solution: By De Morgan's law for universal quantifiers, we know that $\neg\forall x(P(x) \to Q(x))$ and $\exists x(\neg(P(x) \to Q(x)))$ are logically equivalent. By logical equivalence in Table 7 we know that $\neg(P(x) \to Q(x))$ and

Negating Quantified Expressions

When the domain of a predicate $P(x)$ consists of n elements, where n is a positive integer greater than one, the rules for negating quantified statements are exactly the same as De Morgan's laws discussed in Section 1.3. This is why these rules are called De Morgan's laws for quantifiers. When the domain has n elements $x_1, x_2, \ldots, x_n$, it follows that $\neg\forall x P(x)$ is the same as $\neg(P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n))$, which is equivalent to $\neg P(x_1) \vee \neg P(x_2) \vee \cdots \vee \neg P(x_n)$ by De Morgan's laws, and this is the same as $\exists x \neg P(x)$.

Classwork

Show that $\neg\forall x(P(x) \to Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Solution: By De Morgan's law for universal quantifiers, we know that $\neg\forall x(P(x) \to Q(x))$ and $\exists x(\neg(P(x) \to Q(x)))$ are logically equivalent. By logical equivalence in Table 7 we know that $\neg(P(x) \to Q(x))$ and

Example

Use predicates and quantifiers to express the system specifications “Every mail message larger than one megabyte will be compressed” and “If a user is active, at least one network link will be available.”
Solution: Let S(m, y) be “Mail message m is larger than y megabytes,” where the variable x has the domain of all mail messages and the variable y is a positive real number, and let C(m) denote “Mail message m will be compressed.” Then the specification “Every mail message larger than one megabyte will be compressed” can be represented as $\forall m(S(m, 1) \rightarrow C(m))$.
Let $A(u)$ represent “User u is active,” where the variable u has the domain of all users, let $S(n, x)$ denote “Network link n is in state x,” where n has the domain of all network links and x has the domain of all possible states for a network link. Then the specification “If a user is active, at least one network link will be

Quantification of Two Variables

Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

Classwork

Translate the statement “The sum of two positive integers is always positive” into a logical expression.

Solution: To translate this statement into a logical expression, we first rewrite it so that the implied quantifiers and a domain are shown: **“For every two integers, if these integers are both positive, then the sum of these integers is positive.”** Next, we introduce the variables x and y to obtain **“For all positive integers x and y, x + y is positive.”** Consequently, we can express this statement as

$$\forall x \forall y((x > 0) \wedge (y > 0) \rightarrow (x + y > 0)),$$

where the domain for both variables consists of all integers. Note that we could also translate this using the positive integers as the domain. Then the statement “The sum of two positive integers is always positive” becomes **“For every two positive integers, the sum of these integers is positive.”** We can express this as

$$\forall x \forall y(x + y > 0)$$

Classwork

Translate the statement

$$\forall x(C(x) \vee \exists y(C(y) \wedge F(x,y)))$$

into English, where C(x) is “x has a computer,” F(x, y) is “x and y are friends,” and the domain for both x and y consists of all students in your school.

Solution: The statement says that for every student x in your school, x has a computer or there is a student y such that y has a computer and x and y are friends. In other words, **every student in your school has a computer or has a friend who has a computer.**

Classwork

Translate the statement

$$\forall x(C(x) \vee \exists y(C(y) \wedge F(x, y)))$$

into English, where C(x) is “x has a computer,” F(x, y) is “x and y are friends,” and the domain for both x and y consists of all students in your school.

Solution: The statement says that for every student x in your school, x has a computer or there is a student y such that y has a computer and x and y are friends. In other words, **every student in your school has a computer or has a friend who has a computer.**

Classwork

Use quantifiers to express the statement that “There does not exist a woman who has taken a flight on every airline in the world.”

Solution: This statement is the negation of the statement “There is a woman who has taken a flight on every airline in the world”. Our statement can be expressed as

$$\neg\exists w \forall a \exists f (P(w,f) \wedge Q(f,a)),$$

where $P(w,f)$ is “w has taken f ” and $Q(f,a)$ is “f is a flight on a.” By successively applying De Morgan’s laws for quantifiers to move the negation inside successive quantifiers and by applying De Morgan’s law for negating a conjunction in the last step, we find that our statement is equivalent to each of this sequence of statements:

$$\begin{aligned}\forall w \neg\forall a \exists f (P(w,f) \wedge Q(f,a)) &\equiv \forall w \exists a \neg\exists f (P(w,f) \wedge Q(f,a)) \\ &\equiv \forall w \exists a \forall f \neg(P(w,f) \wedge Q(f,a))\end{aligned}$$

Classwork

Use quantifiers to express the statement that “There does not exist a woman who has taken a flight on every airline in the world.”

Solution: This statement is the negation of the statement “There is a woman who has taken a flight on every airline in the world”. Our statement can be expressed as

$$\neg\exists w \forall a \exists f (P(w, f) \wedge Q(f, a)),$$

where $P(w, f)$ is “w has taken f ” and $Q(f, a)$ is “f is a flight on a.” By successively applying De Morgan’s laws for quantifiers to move the negation inside successive quantifiers and by applying De Morgan’s law for negating a conjunction in the last step, we find that our statement is equivalent to each of this sequence of statements:

$$\begin{aligned} \forall w \neg\forall a \exists f (P(w, f) \wedge Q(f, a)) &\equiv \forall w \exists a \neg\exists f (P(w, f) \wedge Q(f, a)) \\ &\equiv \forall w \exists a \forall f \neg(P(w, f) \wedge Q(f, a)) \end{aligned}$$

Rules of Inference

An argument in propositional logic is a sequence of propositions. All but the final proposition in the argument are called **premises** and the final proposition is called the **conclusion**. An argument is **valid** if the truth of all its premises implies that the conclusion is true.

An **argument** form in propositional logic is a sequence of compound propositions involving propositional variables. An **argument form is valid** if no matter which particular propositions are substituted for the propositional variables in its premises, the conclusion is true if the premises are all true.

Rule of Inference	Tautology	Name
P $P \rightarrow Q$ $\therefore Q$	$(P \wedge (P \rightarrow Q)) \rightarrow Q$	Modus Ponens (MP)
$\neg Q$ $P \rightarrow Q$ $\therefore \neg P$	$(\neg Q \wedge (P \rightarrow Q)) \rightarrow \neg P$	Modus Tollens (MT)
$\neg P$ $P \vee Q$ $\therefore Q$	$(\neg P \wedge (P \vee Q)) \rightarrow Q$	Disjunctive Syllogism (DS)
$P \rightarrow Q$ $Q \rightarrow R$ $\therefore P \rightarrow R$	$((P \rightarrow Q) \wedge (Q \rightarrow R)) \rightarrow (P \rightarrow R)$	Hypothetical Syllogism (HS)
Q P $\therefore P \wedge Q$	$(Q \wedge P) \rightarrow (Q \wedge P)$	Conjunction (CONJ)
$P \wedge Q$ $\therefore P$	$(P \wedge Q) \rightarrow P$	Simplification (SIMP)
P $\therefore P \vee Q$	$P \rightarrow (P \vee Q)$	Addition (ADD)
$P \vee Q$ $P \rightarrow R$ $Q \rightarrow R$ $\therefore R$	$((P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow R)) \rightarrow R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ $\therefore Q$	$(P \wedge (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ $\therefore \neg P$	$(\neg Q \wedge (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \vee Q$ $\therefore Q$	$(\neg P \wedge (P \vee Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ $\therefore P \to R$	$((P \to Q) \wedge (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P $\therefore P \wedge Q$	$(Q \wedge P) \to (Q \wedge P)$	Conjunction (CONJ)
$P \wedge Q$ $\therefore P$	$(P \wedge Q) \to P$	Simplification (SIMP)
P $\therefore P \vee Q$	$P \to (P \vee Q)$	Addition (ADD)
$P \vee Q$ $P \to R$ $Q \to R$ $\therefore R$	$((P \vee Q) \wedge (P \to R) \wedge (Q \to R)) \to R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ $\overline{\therefore Q}$	$(P \wedge (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ $\overline{\therefore \neg P}$	$(\neg Q \wedge (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \vee Q$ $\overline{\therefore Q}$	$(\neg P \wedge (P \vee Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ $\overline{\therefore P \to R}$	$((P \to Q) \wedge (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P $\overline{\therefore P \wedge Q}$	$(Q \wedge P) \to (Q \wedge P)$	Conjunction (CONJ)
$P \wedge Q$ $\overline{\therefore P}$	$(P \wedge Q) \to P$	Simplification (SIMP)
P $\overline{\therefore P \vee Q}$	$P \to (P \vee Q)$	Addition (ADD)
$P \vee Q$ $P \to R$ $Q \to R$ $\overline{\therefore R}$	$((P \vee Q) \wedge (P \to R) \wedge (Q \to R)) \to R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ $\therefore Q$	$(P \land (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ $\therefore \neg P$	$(\neg Q \land (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \lor Q$ $\therefore Q$	$(\neg P \land (P \lor Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ $\therefore P \to R$	$((P \to Q) \land (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P $\therefore P \land Q$	$(Q \land P) \to (Q \land P)$	Conjunction (CONJ)
$P \land Q$ $\therefore P$	$(P \land Q) \to P$	Simplification (SIMP)
P $\therefore P \lor Q$	$P \to (P \lor Q)$	Addition (ADD)
$P \lor Q$ $P \to R$ $Q \to R$ $\therefore R$	$((P \lor Q) \land (P \to R) \land (Q \to R)) \to R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ $\therefore Q$	$(P \land (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ $\therefore \neg P$	$(\neg Q \land (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \lor Q$ $\therefore Q$	$(\neg P \land (P \lor Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ $\therefore P \to R$	$((P \to Q) \land (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P $\therefore P \land Q$	$(Q \land P) \to (Q \land P)$	Conjunction (CONJ)
$P \land Q$ $\therefore P$	$(P \land Q) \to P$	Simplification (SIMP)
P $\therefore P \lor Q$	$P \to (P \lor Q)$	Addition (ADD)
$P \lor Q$ $P \to R$ $Q \to R$ $\therefore R$	$((P \lor Q) \land (P \to R) \land (Q \to R)) \to R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ $\therefore Q$	$(P \wedge (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ $\therefore \neg P$	$(\neg Q \wedge (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \vee Q$ $\therefore Q$	$(\neg P \wedge (P \vee Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ $\therefore P \to R$	$((P \to Q) \wedge (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P $\therefore P \wedge Q$	$(Q \wedge P) \to (Q \wedge P)$	Conjunction (CONJ)
$P \wedge Q$ $\therefore P$	$(P \wedge Q) \to P$	Simplification (SIMP)
P $\therefore P \vee Q$	$P \to (P \vee Q)$	Addition (ADD)
$P \vee Q$ $P \to R$ $Q \to R$ $\therefore R$	$((P \vee Q) \wedge (P \to R) \wedge (Q \to R)) \to R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ ——— $\therefore Q$	$(P \land (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ ——— $\therefore \neg P$	$(\neg Q \land (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \lor Q$ ——— $\therefore Q$	$(\neg P \land (P \lor Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ ——— $\therefore P \to R$	$((P \to Q) \land (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P ——— $\therefore P \land Q$	$(Q \land P) \to (Q \land P)$	Conjunction (CONJ)
$P \land Q$ ——— $\therefore P$	$(P \land Q) \to P$	Simplification (SIMP)
P ——— $\therefore P \lor Q$	$P \to (P \lor Q)$	Addition (ADD)
$P \lor Q$ $P \to R$ $Q \to R$ ——— $\therefore R$	$((P \lor Q) \land (P \to R) \land (Q \to R)) \to R$	Disjunction Elimination (DE)

Rule of Inference	Tautology	Name
P $P \to Q$ $\therefore Q$	$(P \wedge (P \to Q)) \to Q$	Modus Ponens (MP)
$\neg Q$ $P \to Q$ $\therefore \neg P$	$(\neg Q \wedge (P \to Q)) \to \neg P$	Modus Tollens (MT)
$\neg P$ $P \vee Q$ $\therefore Q$	$(\neg P \wedge (P \vee Q)) \to Q$	Disjunctive Syllogism (DS)
$P \to Q$ $Q \to R$ $\therefore P \to R$	$((P \to Q) \wedge (Q \to R)) \to (P \to R)$	Hypothetical Syllogism (HS)
Q P $\therefore P \wedge Q$	$(Q \wedge P) \to (Q \wedge P)$	Conjunction (CONJ)
$P \wedge Q$ $\therefore P$	$(P \wedge Q) \to P$	Simplification (SIMP)
P $\therefore P \vee Q$	$P \to (P \vee Q)$	Addition (ADD)
$P \vee Q$ $P \to R$ $Q \to R$ $\therefore R$	$((P \vee Q) \wedge (P \to R) \wedge (Q \to R)) \to R$	Disjunction Elimination (DE)

Classwork

Show that the premises “It is not sunny this afternoon and it is colder than yesterday,” “We will go swimming only if it is sunny,” “If we do not go swimming, then we will take a canoe trip,” and “If we take a canoe trip, then we will be home by sunset” lead to the conclusion “We will be home by sunset.”

Solution: Let p be the proposition “It is sunny this afternoon,” q the proposition “It is colder than yesterday,” r the proposition “We will go swimming,” s the proposition “We will take a canoe trip,” and t the proposition “We will be home by sunset.” Then the premises become

$$\neg p \wedge q, r \rightarrow p, \neg r \rightarrow s, \text{and } s \rightarrow t.$$

The conclusion is simply t. We need to give a valid argument with premises

$$\neg p \wedge q, r \rightarrow p, \neg r \rightarrow s, \text{ and} s \rightarrow t \text{ and conclusion } t.$$

Classwork

Show that the premises "It is not sunny this afternoon and it is colder than yesterday," "We will go swimming only if it is sunny," "If we do not go swimming, then we will take a canoe trip," and "If we take a canoe trip, then we will be home by sunset" lead to the conclusion "We will be home by sunset."

Solution: Let p be the proposition "It is sunny this afternoon," q the proposition "It is colder than yesterday," r the proposition "We will go swimming," s the proposition "We will take a canoe trip," and t the proposition "We will be home by sunset." Then the premises become

$$\neg p \wedge q, r \rightarrow p, \neg r \rightarrow s, \text{and } s \rightarrow t.$$

The conclusion is simply t. We need to give a valid argument with premises

$$\neg p \wedge q, r \rightarrow p, \neg r \rightarrow s, \text{ and } s \rightarrow t \text{ and conclusion } t.$$

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \land q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \to p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \to s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \to t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \land q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \to p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \to s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \to t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Solution in Tabular Form

Step		Reason
1.	$\neg p \wedge q$	Premise
2.	$\neg p$	Simplification using (1)
3.	$r \rightarrow p$	Premise
4.	$\neg r$	Modus tollens using (2) and (3)
5.	$\neg r \rightarrow s$	Premise
6.	s	Modus ponens using (4) and (5)
7.	$s \rightarrow t$	Premise
8.	t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows.

Resolution

Computer programs have been developed to automate the task of reasoning and proving theorems. Many of these programs make use of a rule of inference known as **resolution**. This rule of inference is based on the tautology
$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$.

Example

Use resolution to show that the hypotheses "Jasmine is skiing or it is not snowing" and "It is snowing or Bart is playing hockey" imply that "Jasmine is skiing or Bart is playing hockey."
Solution: Let p be the proposition "It is snowing," q the proposition "Jasmine is skiing," and r the proposition "Bart is playing hockey." We can represent the hypotheses as $\neg p \vee q$ and $p \vee r$, respectively. Using resolution, the proposition $q \vee r$, "**Jasmine is skiing or Bart is playing hockey,**" follows.

Resolution

Computer programs have been developed to automate the task of reasoning and proving theorems. Many of these programs make use of a rule of inference known as **resolution**. This rule of inference is based on the tautology
$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$.

Example

Use resolution to show that the hypotheses "Jasmine is skiing or it is not snowing" and "It is snowing or Bart is playing hockey" imply that "Jasmine is skiing or Bart is playing hockey."

Solution: Let p be the proposition "It is snowing," q the proposition "Jasmine is skiing," and r the proposition "Bart is playing hockey." We can represent the hypotheses as $\neg p \vee q$ and $p \vee r$, respectively. Using resolution, the proposition $q \vee r$, "**Jasmine is skiing or Bart is playing hockey,**" follows.

Resolution

Computer programs have been developed to automate the task of reasoning and proving theorems. Many of these programs make use of a rule of inference known as **resolution**. This rule of inference is based on the tautology
$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$.

Example

Use resolution to show that the hypotheses "Jasmine is skiing or it is not snowing" and "It is snowing or Bart is playing hockey" imply that "Jasmine is skiing or Bart is playing hockey."
Solution: Let p be the proposition "It is snowing," q the proposition "Jasmine is skiing," and r the proposition "Bart is playing hockey." We can represent the hypotheses as $\neg p \vee q$ and $p \vee r$, respectively. Using resolution, the proposition $q \vee r$, "**Jasmine is skiing or Bart is playing hockey,**" follows.

Fallacies

Several common fallacies arise in incorrect arguments. These fallacies resemble rules of inference, but are based on contingencies rather than tautologies.

Example

The proposition $((p \to q) \wedge q) \to p$ is not a tautology, because it is false when p is false and Links q is true. However, there are many incorrect arguments that treat this as a tautology. In other words, they treat the argument with premises $p \to q$ and q and conclusion p as a valid argument form, which it is not. This type of incorrect reasoning is called the fallacy of affirming the conclusion.

Classwork

Is the following argument valid?
If you do every problem in this book, then you will learn discrete mathematics. You learned discrete mathematics.
Therefore, you did every problem in this book.

Solution: Let p be the proposition "You did every problem in this book." Let q be the proposition

"You learned discrete mathematics."

Then this argument is of the form: if $p \to q$ and q, then p. This is an example of an incorrect argument using the fallacy of affirming the conclusion. Indeed, it is possible for you to learn discrete mathematics in some way other than by doing every problem in this book. (You may learn discrete mathematics by reading, listening to lectures, doing some, but not all, the problems in this book, and so on.)

Classwork

Is the following argument valid?
If you do every problem in this book, then you will learn discrete mathematics. You learned discrete mathematics.
Therefore, you did every problem in this book.
Solution: Let p be the proposition "You did every problem in this book." Let q be the proposition

"You learned discrete mathematics."

Then this argument is of the form: if $p \to q$ and q, then p. This is an example of an incorrect argument using the fallacy of affirming the conclusion. Indeed, it is possible for you to learn discrete mathematics in some way other than by doing every problem in this book. (You may learn discrete mathematics by reading, listening to lectures, doing some, but not all, the problems in this book, and so on.)

Rules of Inference for Quantified Statements

Rule of Inference	*Name*
$\dfrac{\forall x P(x)}{\therefore\ P(c)}$	Universal instantiation
$\dfrac{P(c) \text{ for an arbitrary } c}{\therefore\ \forall x P(x)}$	Universal generalization
$\dfrac{\exists x P(x)}{\therefore\ P(c) \text{ for some element } c}$	Existential instantiation
$\dfrac{P(c) \text{ for some element } c}{\therefore\ \exists x P(x)}$	Existential generalization

Universal instantiation

Universal instantiation is the rule of inference used to conclude that P(c) is true, where c is a particular member of the domain, given the premise $\forall x P(x)$. Universal instantiation is used when we conclude from the statement "All women are wise" that "Lisa is wise," where Lisa is a member of the domain of all women.

Universal generalization

Universal generalization is the rule of inference that states that $\forall x P(x)$ is true, given the premise that $P(c)$ is true for all elements c in the domain. Universal generalization is used when we show that $\forall x P(x)$ is true by taking an arbitrary element c from the domain and showing that $P(c)$ is true. The element c that we select must be an arbitrary, and not a specific, element of the domain. That is, when we assert from $\forall x P(x)$ the existence of an element c in the domain, we have no control over c and cannot make any other assumptions about c other than it comes from the domain. Universal generalization is used implicitly in many proofs in mathematics and is seldom mentioned explicitly. However, the error of adding unwarranted assumptions about the arbitrary element c when universal generalization is used is all too common in incorrect reasoning.

Universal instantiation

Universal instantiation is the rule of inference used to conclude that P(c) is true, where c is a particular member of the domain, given the premise $\forall x P(x)$. Universal instantiation is used when we conclude from the statement "All women are wise" that "Lisa is wise," where Lisa is a member of the domain of all women.

Universal generalization

Universal generalization is the rule of inference that states that $\forall x P(x)$ is true, given the premise that $P(c)$ is true for all elements c in the domain. Universal generalization is used when we show that $\forall x P(x)$ is true by taking an arbitrary element c from the domain and showing that $P(c)$ is true. The element c that we select must be an arbitrary, and not a specific, element of the domain. That is, when we assert from $\forall x P(x)$ the existence of an element c in the domain, we have no control over c and cannot make any other assumptions about c other than it comes from the domain. Universal generalization is used implicitly in many proofs in mathematics and is seldom mentioned explicitly. However, the error of adding unwarranted assumptions about the arbitrary element c when universal generalization is used is all too common in incorrect reasoning.

Existential instantiation

Existential instantiation is the rule that allows us to conclude that there is an element c in the domain for which P(c) is true if we know that $\exists xP(x)$ is true. We cannot select an arbitrary value of c here, but rather it must be a c for which P(c) is true. Usually we have no knowledge of what c is, only that it exists. Because it exists, we may give it a name (c) and continue our argument.

Existential generalization

Existential generalization is the rule of inference that is used to conclude that $\exists$ xP(x) is true when a particular element c with P(c) true is known. That is, if we know one element c in the domain for which P(c) is true, then we know that $\exists$ xP(x) is true.

Existential instantiation

Existential instantiation is the rule that allows us to conclude that there is an element c in the domain for which P(c) is true if we know that $\exists xP(x)$ is true. We cannot select an arbitrary value of c here, but rather it must be a c for which P(c) is true. Usually we have no knowledge of what c is, only that it exists. Because it exists, we may give it a name (c) and continue our argument.

Existential generalization

Existential generalization is the rule of inference that is used to conclude that $\exists\, xP(x)$ is true when a particular element c with P(c) true is known. That is, if we know one element c in the domain for which P(c) is true, then we know that $\exists\, xP(x)$ is true.

Classwork

Show that the premises "Everyone in this discrete mathematics class has taken a course in computer science" and "Marla is a student in this class" imply the conclusion "Marla has taken a course in computer science."

Solution: Let $D(x)$ denote "x is in this discrete mathematics class," and let $C(x)$ denote "x has taken a course in computer science." Then the premises are $\forall x(D(x) \rightarrow C(x))$ and D(Marla). The conclusion is C(Marla).

The following steps can be used to establish the conclusion from the premises.

Step		Reason
1.	$\forall x(D(x) \rightarrow C(x))$	Premise
2.	$D(Marla) \rightarrow C(Marla)$	Universal instantiation from (1)
3.	D(Marla)	Premise
4.	C(Marla)	Modus ponens from (2) and (3)

Classwork

Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Marla is a student in this class” imply the conclusion “Marla has taken a course in computer science.”

Solution: Let $D(x)$ denote “x is in this discrete mathematics class,” and let $C(x)$ denote “x has taken a course in computer science.” Then the premises are $\forall x(D(x) \to C(x))$ and D(Marla). The conclusion is C(Marla).

The following steps can be used to establish the conclusion from the premises.

Step		Reason
1.	$\forall x(D(x) \to C(x))$	Premise
2.	$D(Marla) \to C(Marla)$	Universal instantiation from (1)
3.	D(Marla)	Premise
4.	C(Marla)	Modus ponens from (2) and (3)

Classwork

Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Marla is a student in this class” imply the conclusion “Marla has taken a course in computer science.”

Solution: Let $D(x)$ denote “x is in this discrete mathematics class,” and let $C(x)$ denote “x has taken a course in computer science.” Then the premises are $\forall x(D(x) \rightarrow C(x))$ and D(Marla). The conclusion is C(Marla).

The following steps can be used to establish the conclusion from the premises.

Step		Reason
1.	$\forall x(D(x) \rightarrow C(x))$	Premise
2.	$D(Marla) \rightarrow C(Marla)$	Universal instantiation from (1)
3.	D(Marla)	Premise
4.	C(Marla)	Modus ponens from (2) and (3)

Classwork

Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Marla is a student in this class” imply the conclusion “Marla has taken a course in computer science.”

Solution: Let $D(x)$ denote “x is in this discrete mathematics class,” and let $C(x)$ denote “x has taken a course in computer science.” Then the premises are $\forall x(D(x) \to C(x))$ and D(Marla). The conclusion is C(Marla).

The following steps can be used to establish the conclusion from the premises.

Step		Reason
1.	$\forall x(D(x) \to C(x))$	Premise
2.	$D(Marla) \to C(Marla)$	Universal instantiation from (1)
3.	D(Marla)	Premise
4.	C(Marla)	Modus ponens from (2) and (3)

Classwork

Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Marla is a student in this class” imply the conclusion “Marla has taken a course in computer science.”

Solution: Let $D(x)$ denote “x is in this discrete mathematics class,” and let $C(x)$ denote “x has taken a course in computer science.” Then the premises are $\forall x(D(x) \to C(x))$ and D(Marla). The conclusion is C(Marla).

The following steps can be used to establish the conclusion from the premises.

Step		Reason
1.	$\forall x(D(x) \to C(x))$	Premise
2.	$D(Marla) \to C(Marla)$	Universal instantiation from (1)
3.	D(Marla)	Premise
4.	C(Marla)	Modus ponens from (2) and (3)

Classwork

Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Marla is a student in this class” imply the conclusion “Marla has taken a course in computer science.”

Solution: Let $D(x)$ denote “x is in this discrete mathematics class,” and let $C(x)$ denote “x has taken a course in computer science.” Then the premises are $\forall x(D(x) \rightarrow C(x))$ and D(Marla). The conclusion is C(Marla).

The following steps can be used to establish the conclusion from the premises.

Step		Reason
1.	$\forall x(D(x) \rightarrow C(x))$	Premise
2.	$D(Marla) \rightarrow C(Marla)$	Universal instantiation from (1)
3.	D(Marla)	Premise
4.	C(Marla)	Modus ponens from (2) and (3)

References

[1] K. H. Rosen. *Discrete mathematics & applications*. McGraw-Hill, 2018.